Amendments to the Claims

This listing of claims will replace all the prior revisions, and listings of claims in this application.

Listing of Claims

1. (Withdrawn) A method for distributing data between a source and destination in a network, the method comprising the steps of:

defining, n number of paths between the source and the destination;

splitting the data into n number of blocks

independently transmitting the n blocks from the source to the destination; and

reassembling, at the destination, the n blocks into the data.

2. (Withdrawn) The method according to claim 1 wherein the independent transmission of the n blocks is not started simultaneously.

3. (Withdrawn) The method according to claim 1 wherein the size of the n blocks remains constant for the duration of the transmission.

4. (Withdrawn) The method according to claim 1 wherein the size of the n blocks varies during the duration of the transmission.

5. (Withdrawn) The method according to claim 1 wherein the n number of paths is determined using multicast, depth-first-search technique.

6. (Withdrawn) The method according to claim 1 wherein the n number of paths is determined using multicast spanning tree technique.

7. (Withdrawn) The method according to claim 1 wherein the n number of paths is determined using multicast arborescence technique.

8. (Withdrawn) The method according to claim 1 further comprising the step of:

receiving, at the source, a request for a data transfer to the destination.

9. (Currently Amended) A method of file transfer in a computer-based communication network by utilizing an overlay network composed of cooperating servers on computer hosts, said computer hosts connected to said communication network, wherein each said server contains instructions which, when executed by said server, cause said server to process and forward data via the transport layer to other servers on said overlay network without modifying the native data transport protocol at transport or lower layers, said method comprising the steps of:

defining a first data forwarding path between two servers, said path comprising of concatenation of overlay links, each of said links established via transport layer between the two said servers in said overlay network;

defining a second data forwarding path between the two servers, said path different from the first data forwarding path;

dividing the data file in at least two sub-files, first sub-file and second sub-file; and

sending first and second sub-files over the first and second data forwarding paths, respectively; and

assembling the first and second sub-files into a single file at the second server.

10. (Currently Amended) The method according to claim 9 further comprising the steps of:

replicating each sub-file in one or more intermediate servers on said data forwarding paths and forwarding the sub-file to the next server in the path~~; and~~

~~assembling the first and the second sub-files at the second server~~.

11. (Original) The method of claim 9, wherein said sending first and second sub-files over the first and second data forwarding paths are not started simultaneously.

12. (Original) The method of claim 10, wherein said sub-files are transmitted from a single source node to a plurality of destination nodes, and said paths form a data forwarding tree, wherein the intermediate nodes of the tree copy the sub-files reaching them from the incoming overlay link to the plurality of outgoing overlay links.

13. (Original) The method of claim 9, wherein said sub-files are transmitted from a plurality of source nodes to a plurality of destination nodes, and said forwarding paths form data forwarding trees, wherein the intermediate nodes of the tree copy the data reaching them from the incoming link to the plurality of outgoing links.

14. (Original) The method of claim 9, wherein the data forwarding and processing resources are reserved, said resources including one or more metrics from the following group: bandwidth of overlay link, processing load of the server.

15. (Original) The method of claim 9, wherein the data forwarding and processing resources are determined from network monitoring, said resources including one or more metrics from the following group: bandwidth of overlay link, processing load of the server.

16. (Original) The method of claim 9, wherein the said overlay network has static topology and resources, said resources including one or more metrics from the following group: bandwidth, processing load.

17. (Original) The method of claim 9, wherein the said overlay network has dynamic topology and resources, said resources including one or more metrics from the following group: bandwidth, processing load.

18. (Original) The method of claim 9, wherein the data forwarding paths are computed in a single server.

19. (Original) The method of claim 9, wherein the data forwarding paths are computed in several servers, with subsequent coordination of computed results.

20. (Original) The method of claim 9, wherein the divisions of the data file into sub-files remain constant for the duration of the file transfer.

21. (Original) The method of claim 9, wherein the divisions of the data file into sub-files change during the file transfer.

22.(Original) The method of claim 11, wherein the data forwarding tree is constructed using multicast depth-first-search method.

23. (Original) The method of claim 11, wherein the data forwarding tree is constructed using multicast spanning tree method.

24. (Original) The method of claim 11, wherein the data forwarding tree is constructed using multicast arborescence method.

25. (Original) The method of claim 9, wherein the established communications protocols include one or more protocols selected from the following group: Internet Protocol, http, ftp SSL, TCP reliable UDP using erasure coding.

26. (Currently Amended) The method of assembly of claim 12, wherein the assembly is done at overlay network layer, said method comprising the steps of:

encoding a monotonically increased sequence number in each packet header at data source.

queuing data packet in a sink buffer in destination for each data transport session.

scanning a sink buffer and selecting a data packet that matches the current receiving sequence number maintained at the destination.

clocking out available data packets in the sink buffer selected by ~~step c)~~ the queuing step above;

increasing the receiving sequence number at the destination.

delivering the available data packet to the application client; and

repeating the method for all sink buffers.

27. (Original) The method of assembly of claim 12, wherein the assembly is done at the overlay application layer by the application, said method comprising the steps of:

encoding an application-specific data object identifier into a data frame;

extracting the application-specific data object identifier at the destination; and

reassembling the application data object according to the data object identifier.

28. (Currently Amended) The method of data transfer of claim 9, said method comprising the steps of:

using an explicit or implicit method to setup the forwarding table in each overlay node;

forwarding the data by looking up the forwarding table installed by the setup step above ~~step a)~~ at each node;

scheduling data transport at the data source and at each intermediate node for each path or tree according to their QoS specifications; and

slowing down a particular data flow on a path or tree by using backpressure.

29. (Currently Amended) The method of claim 9, wherein the transport of data partitions to a plurality of destination nodes is a combined coordination of transport push initiated by the sender ~~(push)~~ and transport pull initiated by the receivers ~~(pull)~~.

30. (Cancelled).